robotron

Anwendungsbeschreibung

Betriebssystem SCP 1700

C 1012-0000-1 M 3030

Arbeitsplatzcomputer A 7100 Betriebssystem SCP 1700

Anwendungsbeschreibung

Betriebssystem SCP 1700

AC A7100

VEB Robotron-Projekt Dresden

C 1012-0000-1 M 3030

Die vorliegende Systemunterlagendokumentation, Anwendungsbeschreibung Betriebssystem SCP 1700 , entspricht dem Stand von 6/86.

Nachdruck, jegliche Vervielfältigung oder Auszüge daraus sind unzulässig.

Die Ausarbeitung erfolgte durch ein Kollektiv des VEB Robotron-Elektronik Dresden .

Im Interesse einer ständigen Weiterentwicklung werden alle Leser gebeten, Hinweise zur Verbesserung dem Herausgeber mitzuteilen.

Herausgeber:

VEB Robotron-Projekt Dresden 8010 Dresden, Leningrader Str. 9

(C) VEB Robotron 1986

Kurzreferat

Diese Schrift gibt einen Überblick über das Betriebssystem SCP 1700. Sie enthält Angaben zu den Bestandteilen des Betriebssystems, den gerätetechnischen Voraussetzungen und zu möglichen Einsatzgebieten sowie eine Kurzbeschreibung der Arbeitsweise des SCP-1700-Steuerprogramms, der Dienstprogramme und Übersetzer.

Die Schrift sollte von allen Personen gelesen werden, die sich mit der Planung und Vorbereitung des Einsatzes des Arbeitsplatzcomputersystems A7100 befassen. Sie sollte weiterhin von Nutzern des Betriebssystems SCP 1700 als erste einführende Schrift gelesen werden.

Inhaltsverzeichnis Seite

1.	Einführung	5
2.	Aufgabenstellung	6
2. 1.	Allgemeine Gesichtspunkte	6
2. 2.	Einsatzgebiet	6
2. 3.	Bestandteile des BS SCP 1700	7
2. 4.	Speicherbedarf für das BS SCP 1700	9
3.	Gerätekonfiguration	10
4.	Steuerprogramm SCPX 1700	11
4. 1.	Bestandteile des Steuerprogramms	11
4. 2.	Speicherverwaltung	12
4. 3.	Datenträgerorganisation	12
4. 4.	SCP-1700-Kommandos	13
5.	Dienstprogramme	15
5. 1.	Assembler ASM86	15
5. 2.	Ladeformatgenerierungsprogramm GENCMD	15
5. 3.	Dateitransferprogramm PIP	16
5. 4.	Editor ED	16
5. 5.	Testhilfe DDT86	17
5. 6.	Systemstatusprogramm STAT	18
5. 7.	Stapelverarbeitungsprogramm SUBMIT	18
5. 8.	Diskettenkopierprogramm COPYDISK	18
5. 9.	Diskettenformatierprogramm INIT	19
5. 10.	LDSCP-Kopierprogramm LDCOPY	19
5. 11.	Bedienerhilfsprogramm HELP	19
5. 12.	Programm zur Manipulierung des Diskettenzugriffs DISKSET	20
5. 13.	Steuertabellengenerierungsprogramm GENDEF	20
6.	Übersetzer	21
6. 1.	BASIC-Interpreter BASIC	21
6. 2.	FORTRAN-Compiler FOR77	21
7.	Programmpaket für modulare Programmierung	23
7. 1.	Assembler für linkbare Programme RASM86	23
7. 2.	Ausgabeprogramm für Cross-Referenzlisten XREF86	23
7. 3.	Bibliothekar LIB86	24
7. 4.	Linker LINK86	24
7. 5.	Symbolischer Debugger SID86	25
8.	SCP-1700-Anpassung	26
Anlage 1	Typenbezeichnung der Geräte	27
Anlage 2	Abkürzungsverzeichnis	28
Sachwortverzeichnis		29

1. Einführung

In der vorliegenden Schrift werden Funktion und Zusammensetzung des Betriebssystems SCP 1700 beschrieben. Damit soll allen Personen, deren Tätigkeit mit dem Arbeitsplatzcomputer A7100 verbunden ist, ein Überblick über das Betriebssystem SCP 1700 gegeben werden. Zum Verständnis der vorliegenden Schrift sind Grundkenntnisse auf dem Gebiet der elektronischen Datenverarbeitung erforderlich. Darüber hinaus ist die Kenntnis der Übersichtsabschitte der Betriebsdokumentation des Arbeitsplatzcomputers A7100 nützlich.

Die Anwenderdokumentation des Betriebssystems SCP 1700 besteht aus den Teilen

- Programmtechnische Beschreibungen für SCP 1700 bestehend aus den Teilen:
  - Executive SCPX 1700
    Anleitung für den Systemprogrammierer
  - Assembler und Debugger
    Anleitung für den Programmierer und
    Anleitung für den Bediener
  - Dienstprogramme
    Anleitung für den Bediener

- Programmpaket für modulare Programmierung
  Anleitung für den Bediener und
  Anleitung für den Programmierer

- BASIC-Interpreter
  Programmtechnische Beschreibung

- FORTRAN-77
  Sprachbeschreibung

- FORTRAN-77 Benutzungshinweise
  Anleitung für den Programmierer

- Symbolischer Debugger SID86
  Anleitung für den Bediener

2. Aufgabenstellung

2.1. Allgemeine Gesichtspunkte

Das hier vorgestellte Betriebssystem SCP 1700 ist Bestandteil der Softwareausstattung für den vom VEB Kombinat Robotron konzipierten Arbeitsplatzcomputer A7100.

Auf Grund seiner guten technischen Eigenschaften, insbesondere der 16-bit-Verarbeitungsbreite und der relativ großen Hauptspeicherkapazität ist der A7100 sehr vielseitig einsetzbar. Er ist unter anderem für den Einsatz im kommerziellen Bereich, als Progammentwicklungssystem, in der Labor- und Prüffeldautomatisierung, in der Konstruktion und Technologie und in der Meßtechnik als Auswerterechner geeignet.

Diese unterschiedlichen Einsatzgebiete stellen auch differenzierte Anforderungen an die jeweils verwendeten Betriebssysteme des Rechners. Das Betriebssystem SCP 1700 soll in erster Linie die Anforderungen abdecken, die sich aus dem Einsatz des A7100 im kommerziellen Bereich ergeben. Es ist nicht geeignet für Echtzeitanwendungen.

Außer der unmittelbaren Unterstützung der Anwendung des Rechners A7100 werden mit dem Betriebssystem SCP 1700 technologische Mittel zum Erstellen und Austesten von Anwenderprogrammen bereitgestellt. Die Entwicklung des Betriebssystems SCP 1700 erfolgte mit dem Ziel, auf dem Niveau der Anwenderprogramme Kompatibilität zu dem im NSW eingesetzten Betriebssystem CP/M-86 zu gewährleisten.

Zu dem im Inland bereits vorhandenen Betriebssystem SCP 1520 für die Bürocomputer A5120/30 und PC1715 besteht Kompatibilität hinsichtlich der unterstützten Diskettenformate und des Dateiaufbaus sowie der Programmiersprache BASIC.

2.2. Einsatzgebiet

Das Betriebssystem SCP 1700 ist ein externspeicherorientiertes Einzelnutzerbetriebssystem, das in erster Linie für den Einsatz des A7100 als Büro- und Personalcomputer vorgesehen ist. Typische Anwendungen mit SCP 1700 als Basisbetriebssystem sind z. B.

- Buchung
- Fakturierung - Abrechnung
- wiss.-techn. Berechnungen
- Textverarbeitung

SCP 1700 eignet sich außerdem zur Software-Entwicklung.

2.3. Bestandteile des BS SCP 1700

Das Betriebssystem SCP 1700 enthält folgende Programme

- Steuerprogrammlader (LDSCP)

- Steuerprogramm (SCPX)

- Dienstprogramme (Kommandos)
  - Assembler (ASM86)
  - Ladeformatgenerierungsprogramm (GENCMD)
  - Dateitransferprogramm (PIP)
  - Editor (ED)
  - Testhilfe (DDT86)
  - Steuertabellengenerierungsprogramm (GENDEF)
  - Systemstatusprogramm (STAT)
  - Stapelverarbeitungsprogramm (SUBMIT)
  - Diskettenkopierprogramm (COPYDISK)
  - Diskettenformatierprogramm (INIT)
  - Bedienerhilfsprogramm (HELP)
  - LDSCP-Kopierprogramm (LDCOPY)
  - Programm zur Manipulierung des Diskettenzugriffs (DISKSET)

- Übersetzer
  - BASIC-Interpreter (BASIC)
  - Fortran-Compiler (FOR77)

- Programmpaket für modulare Programmierung
  - Assembler für linkbare Programme (RASM86)
  - Ausgabeprogramm für Cross-Referenzlisten (XREF86)
  - Linker (LINK86)
  - Bibliothekar (LIB86)
  - Symbolischer Debugger (SID86)

Das Betriebssystem SCP 1700 besteht demnach aus dem Steuerprogrammlader, dem Steuerprogramm und einer Reihe von Systemprogrammen (Transientkommandos). Der Steuerprogrammlader wird nach Einschalten des A7100 durch das A7100-Monitorprogramm von den ersten zwei Spuren der Systemdiskette in den Speicher geladen und gestartet. Er lädt dann das SCP-1700-Steuerprogramm in den Hauptspeicher und übergibt ihm die Steuerung.

Die Systemprogramme arbeiten unter Regie des Steuerprogramms. Sie dienen der Programmentwicklung und -übersetzung, der Ausführung spezieller Systemverwaltungsfunktionen und der Arbeit mit Dateien.

Das Programmpaket für modulare Programmierung unterstützt die Assemblierung getrennt übersetzbarer Programmeinheiten, sowie deren Zusammenfügen zu einem Programmsystem unter eventueller Hinzunahme von weiteren Moduln aus Objektbibliotheken. Für die Arbeit mit FORTRAN und später hinzukommenden Compilern ist das Programmpaket Voraussetzung.

Das Zusammenwirken verschiedener Systemprogramme zur Herstellung eines lauffähigen Programms zeigt das folgende Bild am Beispiel eines in der Assemblersprache ASM86 geschriebenen Programms:

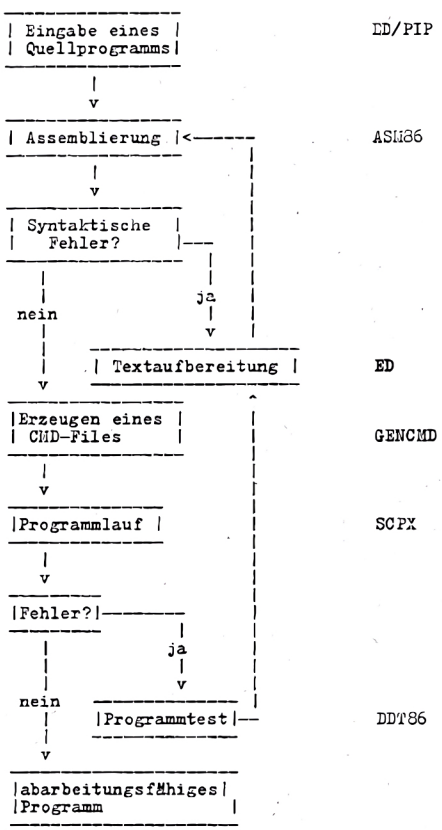

Bild: Herstellung eines lauffähigen Programms

2.4 Speicherbedarf für das BS SCP 1700

Der haupspeicherresidente Teil des Betriebssystems (Steuerprogramm SCPX 1700) belegt ca. 15K Byte im Hauptspeicher. Um den Assembler ASM86 und die SCP-1700-Dienstprogramme abzuarbeiten, werden 64K Byte benötigt.

3. Gerätekonfiguration

Das Betriebssystem SCP 1700 ist für die Arbeit mit dem AC A7100 vorgesehen. Sein Steuerprogramm SCPX 1700 unterstützt die folgenden Geräte und Geräteschnittstellen:

Logische Geräte	Physische Geräte
- CONSOLE	Tastatur und Bildschirmeinheit oder Bildschirmterminal (IFSS oder V24)
- LIST	Seriendrucker (Centronics, IFSS oder IFSP)
- AXI	Eingabegerät (IFSS oder V24)
- AXO	Ausgabegerät (IFSS, IFSP oder V24)
- Externspeicher	2 Minifolienspeicherlaufwerke im Grundgefäß 2 Minifolienspeicherlaufwerke oder bis zu 2 Standardfolienspeicherlaufwerke in Beistellgefäßen

Die Typenbezeichnung der Geräte entsprechend dem gegenwärtigen Entwicklungsstand sind in der Anlage enthalten.
Als Minimalperipherie sind Tastatur, Bildschirmeinheit und Minifolienspeichereinheit erforderlich. Für den Operationsspeicher ist die im AC kleinstmögliche Größe von 128K Byte ausreichend.

4. Steuerprogramm SCPX 1700

4.1. Bestandteile des Steuerprogramms

Das Steuerprogramm SCPX 1700 ist der hauptspeicherresidente Teil des Betriebssystems SCP 1700. Er besteht aus drei Moduln:

- CCP (Command Console Processor)
- BDOS (Basic Disk Operating System)
- BIOS (Basic I/O System)

CCP

CCP ist ein Programm, das einerseits den Nutzerdialog führt, andererseits durch den Nutzer aufgerufene Programme sucht und von der Diskette in den Transientbereich (vom Betriebssystem nicht belegter Teil des Hauptspeichers) lädt und startet. Alle abarbeitbaren Programme (Transientkommandos) stehen als Dateien des Typs CMD auf Diskette. Fünf Grundkommandos sind im CCP fest eingebaut (built-in) und damit hauptspeicherresident.

BDOS

BDOS enthält 51 Betriebssystemfunktionen für einfache E/A, Dateioperationen (sequentieller und direkter Zugriff), Speicherverwaltung und Programmladen. Es wird eine einheitliche Rufkonvention realisiert, wobei der Eintritt in das BDOS stets über den Softwareinterrupt 224 erfolgt. Der Aufruf der Betriebssystemfunktionen wird über einen Funktionscode realisiert, der zusammen mit den Parametern in Registern übergeben wird. Die Rückkehrwerte stehen ebenfalls in Registern. Mit der Funktion 0 kann eine Rückgabe der Steuerung an den CCP erfolgen.

BIOS

BIOS enthält die hardwareabhängigen Betriebssystemteile. Durch Modifizierung des BIOS kann SCPX 1700 auf eine spezifische Hardware-Umgebung zugeschnitten werden.
Der Eintritt in das BIOS erfolgt über BDOS-Rufe. BIOS enthält 21 Subroutinen für die Systeminitialisierung, einfache Zeichen-E/A und Disketten-E/A.
Periphere Geräte werden von SCP 1700 als logische Geräte betrachtet, die Zuordnung zu physischen Geräten erfolgt innerhalb des BIOS.
Logische Geräte sind die Geräte CONSOLE, LIST, READER und PUNCH sowie maximal 16 Externspeichereinheiten (A, B ... P).

4.2. Speicherverwaltung

Der in der vorliegenden Hardware-Konfiguration vorhandene und nicht vom Steuerprogramm belegte RAM-Bereich ist in einer Tabelle im BIOS erfaßt. Der RAM-Bereich muß nicht unbedingt zusammenhängend sein. Die Tabelle kann Einträge über maximal 8 Teilbereiche enthalten.

Über diesen statisch erfaßten Speicher realisiert SCPX 1700 eine dynamische Speicherverwaltung. Der verfügbare Speicher kann in 8 Regionen aufgeteilt werden. Eine Anforderung zur Reservierung einer Speicherregion erfolgt entweder implizit beim Laden eines Transientprogramms durch den CCP oder explizit durch ein Nutzerprogramm mittels entsprechender BDOS-Rufe.
Die dynamische Speicherverwaltung erlaubt dem Nutzer zum Beispiel mehrere Programme in den Hauptspeicher zu laden.

4.3. Datenträgerorganisation

Diskettenformate

Auf den am AC A7100 anschließbaren Diskettenlaufwerken sind folgende Disketten einzusetzen:

K5602.10 8"-Diskette, einseitig, einfache Dichte, 77 Spuren
K5600.20 5,25"-Diskette, einseitig, doppelte Dichte, 80 Spuren

Das physische Datenformat der Disketten entspricht bzw. lehnt sich an die Standards
KROS 5108 (für 8"-Disketten) und
KROS 5110 (für 5,25"-Disketten) an.
Die Disketten werden mit dem Programm INIT 1700 formatiert. Standardmäßig wird bei SCP 1700 mit Hausformaten gearbeitet, die von den SCP-1520-Formaten abgeleitet werden. Für 8"-Disketten ist außerdem das CP/M-Standardformat realisiert.

Diskette	Physisches Format	Formatbezeichnung Kapazität für Anwender
8"-Diskette einseitig einfache Dichte	Spur 0 : 26 Sektoren a 128 Byte Spur 1-76: 4 Sektoren a 1024 Byte	Hausformat/ 294K Byte
oder	Spur 0-76: 26 Sektoren a 128 Byte	CP/M-Standardformat/ 239K Byte
5,25"-Diskette einseitig, doppelte Dichte	Spur 0(FM);16 Sektoren a 128 Byte Spur 1-79: 16 Sektoren a 256 Byte	Hausformat/ 306K Byte

Diskettenverwaltung

Der Folienspeicher wird durch SCPX 1700 dynamisch verwaltet. Beim allerersten Zugriff auf ein Laufwerk durchsucht SCPX 1700 das Verzeichnis der dort vorhandenen Diskette und legt eine Tabelle darüber an, welche Abschnitte der Diskette mit gültigen Aufzeichnungen belegt und welche noch frei sind. Diese Tabelle dient als Grundlage zur Berechnung des freien Speicherplatzes und zur Auswahl neuer Aufzeichnungsabschnitte. Zusätzlich wird eine Prüfsumme über den Inhalt jedes Verzeichnisabschnitts berechnet und in einer besonderen Prüftabelle abgelegt. Anhand dieser Prüftabelle kann SCPX 1700 beim Zugriff auf ein Laufwerk feststellen, ob ein Diskettenwechsel stattgefunden hat.

Dateiverwaltung

Der Zugriff zu Dateien erfolgt über ein Dateiverzeichnis. Jeder Verzeichniseintrag enthält außer dem Dateinamen Angaben über die durch die Datei belegten Speicherblöcke; über die Zugehörigkeit zu einer bestimmten Nutzergruppe sowie über bestimmte Dateiattribute (z. B. Nur-Lese-Dateien).

Die Nutzergruppe wird durch Zuweisung einer Nutzernummer beim Anlegen einer Datei festgelegt. Alle Systemprogramme, außer PIP, haben nur zu den Dateien Zugriff, die die Nutzernummer besitzen, unter der augenblicklich gearbeitet wird.

Die maximale Dateigröße beträgt 8 Megabyte.

4.4. SCP-1700-Kommandos

Kommandotypen

SCP 1700 meldet seine Bereitschaft zur Entgegennahme eines Kommandos durch Ausgabe der Zeichenfolge

d>

d steht für den Namen des Bezugslaufwerkes (A ... P), d. h. desjenigen Laufwerkes, auf den sich alle Dateinamen in der Kommandozeile beziehen, wenn sie keine explizite Laufwerksangabe enthalten.

SCP 1700 kennt zwei Arten von Kommandos:
- ins Betriebssystem (CCP) selbst eingebaute residente Kommandos
- transiente Kommandos, die als Dateien vom Typ CMD auf der Diskette stehen und bei Aufruf in den Operationsspeicher geladen werden

Jeder Anwender kann eigene Transientkommandos hinzufügen, indem er selbst Programme schreibt, übersetzt und als CMD-Dateien auf einer Diskette bereitstellt. Anwenderprogramme werden also vom Betriebssystem SCP 1700 wie Transientkommandos behandelt.

Residente Kommandos

DIR/DIRS	Ausgabe von Datenträgerverzeichnissen
ERA	Löschen von Dateien
REN	Umbenennen von Dateien
TYPE	Anzeigen des Inhalts von ASCII-Dateien
USER	Ändern des Nutzerbereiches
d:	Bezugslaufwerk umschalten (d ... Laufwerksbezeichnung: A ... P)

Steuerzeichenkommandos

CTRL/C Beenden des aktuellen Programms bzw. bei Eingabe nach Kommandoanforderungen des CCP Reininitialisierung des Systems

CTRL/P Parallelschalten des Druckers zur Konsolenausgabe bis zum nächsten CTRL/P

CTRL/S Anhalten einer laufenden Konsolenausgabe bis zum nächsten Tastendruck

Transientkommandos

Transientkommandos sind lauffähige Programme, die als Dateien des Typs CMD auf der Diskette stehen (Dienstprogramme, Assembler, Compiler, Nutzerprogramme). Beim Laden eines Transientkommandos ermittelt die Laderfunktion des SCPX 1700 das Speichermodell, für welches das zu ladende Programm entwickelt wurde und initialisiert dementsprechend die Segmentregister. Drei Speichermodelle werden unterschieden:

8080-Modell	Das Programm besteht aus einer Code-Gruppe, die Code und Daten gemischt enthält (max. 64K Byte)
Kleines Modell (small model)	Das Programm besteht aus einer Code- und einer Daten-Gruppe
Kompaktes Modell (compact model)	Das Programm besteht aus drei oder mehr unabhängigen Gruppen

SCPX 1700 initialisiert während des Ladens einen speziellen Bereich im Programm, die sogenannte Basisseite (base page), mit Informationen über Länge und Basisadresse der im Programm enthaltenen Gruppen, sowie über die Kommandozeile zum Aufruf des Transientkommandos.

5. Dienstprogramme

5.1. Assembler ASM86

Aufgaben

Der Assembler übersetzt ASM86-Quellprogramme in Objektprogramme im hexadezimalen Format, die durch GENCMD weiterverarbeitet werden können.

Arbeitsweise

Der Assembler arbeitet in drei Durchläufen und erzeugt außer der hexadezimalen Datei ein Assemblerprotokoll sowie eine Liste der im Quellprogramm verwendeten Symbole. Durch Optionen kann der Nutzer steuern, welche Geräte für welche Dateien benutzt und ob Ausgabedateien unterdrückt werden sollen.

ASM86 bietet Direktiven zur
- Programmsegmentierung (absolute und verschiebliche Segmente)
- Manipulation des Speicherplatzzählers
- bedingten Assemblierung
- Definition von Symbolen
- Speicherplatzreservierung und -initialisierung
- Einbeziehung von INCLUDE-Dateien
- Listensteuerung

Außerdem kann der Nutzer durch Verwendung von Code-Makros eigene Maschinenbefehle definieren.

5.2. Ladeformatgenerierungsprogramm GENCMD

Aufgaben

GENCMD transformiert ein Objektprogramm aus dem durch den Assembler ASM86 erzeugten hexadezimalen Format (H86) in das SCP-1700-Ladeformat (CMD).

Arbeitsweise

Die durch GENCMD erzeugte CMD-Datei besteht aus einem 128 Byte langen Vorsatz, gefolgt vom Speicherabbild des Programms. Der Vorsatz beinhaltet alle notwendigen Informationen über die im Speicherabbild enthaltenen Gruppentypen (siehe Abschnitt 4.4. Transientkommandos). Gewonnen werden diese Informationen zum einen aus speziellen Sätzen der hexadezimalen Datei und zum anderen aus Optionen beim Aufruf von GENCMD.

5.3. Dateitransferprogramm PIP

Aufgaben

PIP ist ein Programm zum Informationsaustausch zwischen Peripherieeinheiten, wobei unter Peripherieeinheiten Folienspeicher und die über die Ein-/Ausgabekanäle CONSOLE, READER, PUNCH und LIST angeschlossenen Geräte zu verstehen sind. Mit Hilfe von PIP lassen sich Daten von einer derartigen Peripherieeinheit zu einer anderen übertragen. Beispielsweise sind möglich:

- Kopieren einer Datei in eine andere
- Kopieren ganzer Disketten
- Ausdrucken des Inhalts einer Textdatei
- Auslisten des Inhalts einer Datei auf den Bildschirm
- Zusammenfügen mehrerer Einzeldateien zu einer Gesamtdatei
- Erstellen von Dateiauszügen
- Umformen von Textdateien (Groß- in Kleinschreibung, Verändern der Seitenlänge, Verändern der Zeilenlänge u. ä.)

Arbeitsweise

Die Funktionen von PIP werden über Optionen in der Kommandozeile aufgerufen.

5.4. Editor ED

Aufgaben

Der Editor ED ist ein Programm zum Anlegen und Verändern von Textdateien. Mit seiner Hilfe lassen sich
- Textdateien erzeugen
- Texte in vorhandene Dateien einfügen
- Texte in vorhandenen Dateien löschen
- Texte in vorhandenen Dateien umgruppieren
  und mit Einschränkungen auch
- Textausschnitte erstellen
- vorhandene Textdateien zusammenfügen

Arbeitsweise

Durch ED wird der Text aus der Originaldatei über einen Pufferspeicher in eine Zwischendatei kopiert. Dieser Kopiervorgang kann durch ED-Kommandos interaktiv gesteuert werden.

Es kann
- eine gewünschte Anzahl von Zeilen in den Puffer eingelesen werden
- eine gewünschte Anzahl von Zeilen in die Zwischendatei geschrieben werden
- Text, solange er im Pufferspeicher ist, wie folgt bearbeitet werden:
  - neuen Text über die Tastatur oder von einer Datei aus einfügen
  - Textausschnitte in eine Datei schreiben
  - beliebige Textstellen aufsuchen
  - beliebige Textstellen löschen
  - beliebige Textstellen verändern

Wenn der Nutzer den Editiervorgang beendet, überträgt ED den Inhalt des Puffers, gefolgt von dem noch unbearbeiteten Text aus der Quelldatei in die Zwischendatei. Anschließend werden die Dateien umbenannt. Die Quelldatei, deren Inhalt während des Editiervorgangs nicht verändert wird, bleibt dem Nutzer unter ihrem ursprünglichen Dateinamen, jedoch mit dem Dateityp BAK erhalten. Die Zwischendatei erhält den Dateinamen sowie den Dateityp der Quelldatei.

5.5. Testhilfe DDT86

Aufgaben

DDT86 dient zum Testen von Nutzerprogrammen, die im CMD-Format auf Diskette bereitstehen müssen.
Mit DDT86 werden in erster Linie Programe getestet, die in Assemblersprache geschrieben wurden.

Arbeitsweise

Nutzerprogramme können entweder zusammen mit DDT86 oder durch DDT86-Kommandos geladen werden. Über Bedienerkommunikation können folgende Leistungen von DDT86 angefordert werden:

- Anzeige und Änderung von Register- und Speicherinhalten
- Auslisten von Speicherinhalten in mnemonischer Form
- Eingabe von Assembler-Anweisungen in mnemonischer Form
- Vergleichen von Speicherblöcken
- Verschieben von Speicherblöcken
- Berechnung der Summe und Differenz zweier Hexadezimalzahlen
- Ein- und Ausgaben von Dateien in bzw. aus dem Speicher (für Patching)
- Steuerung der Abarbeitung des Nutzerprogramms mit und ohne Protokollierung (schrittweise Abarbeitung, Setzen von Unterbrechungspunkten)

Die schrittweise Befehlsabarbeitung erfolgt unter Ausnutzung des Single-Step-Interrupts. BDOS-Rufe werden im Single-Step-Modus als ein Befehl behandelt, da DDT86 selbst BDOS-Rufe benutzt und BDOS nicht reentrant ist. Unterbrechungspunkte werden realisiert, indem DDT86 auf die vom Nutzer festgelegte Programmadresse den Code für den One-Byte-Interrupt (INT 3) einträgt. DDT86 sichert den ursprünglichen Inhalt der Unterbrechungsadresse und stellt diesen Inhalt nach einer Unterbrechung wieder her.

5.6. Systemstatusprogramm STAT

Aufgabe

STAT dient dazu, sich einen Überblick über die wichtigsten Systemeigenschaften zu verschaffen und diese, soweit möglich, gezielt zu verändern. Dies betrifft sowohl die Peripheriegeräte (Ein- und Ausgabegeräte, Massenspeichereinheiten) als auch Speicherbedarf und Merkmale von Dateien.

Arbeitsweise

Die verschiedenen Dienste werden durch Angabe von Optionen in der Kommandozeile zur Aktivierung von STAT angefordert.
Durch STAT VAL: werden dem Nutzer alle möglichen Optionen von STAT ausgelistet. Folgende Dienste werden geboten:
- Ausgabe einer Übersicht über die auf einer Diskette vorliegenden Benutzerbereiche (Nutzernummern)
- Auslisten der aktuellen Zuordnung von E/A-Kanälen und Geräten
- Zuweisung von Geräten zu E/A-Kanälen
- Information über den freien Speicherplatz auf einer Diskette
- Auslisten der Merkmale und des belegten Speicherplatzes von Dateien
- Setzen von Dateimerkmalen
- Auslisten der Eigenschaften der derzeit aktiven Disketten

5.7. Stapelverarbeitungsprogramm SUBMIT

Aufgaben

Durch SUBMIT hat der Nutzer die Möglichkeit, Kommandodateien zur automatischen Abarbeitung an SCPX 1700 zu übergeben.

Arbeitsweise

Eine Kommandodatei ist eine Folge von gültigen SCP-1700-Kommandos. Sie muß den Dateityp SUB haben. Die Kommandos können formale Parameter enthalten, die beim Aufruf von SUBMIT durch in der Kommandozeile anzugebende aktuelle Parameter ersetzt werden. Das SUBMIT-Kommando selbst kann in Kommandodateien nur als letztes Kommando stehen.

5.8. Diskettenkopierprogramm COPYDISK

Aufgaben

COPYDISK kopiert alle Informationen einer Diskette auf eine andere einschließlich der SCP-1700-Systemspuren, falls sie auf der Quelldiskette vorhanden sind.

Arbeitsweise

Der Kopiervorgang erfolgt spurweise. Voraussetzung für die Arbeit von COPYDISK ist, daß die Zieldiskette formatiert ist und die gleichen Parameter wie die Quelldiskette aufweist.
Die Formatierung der Zieldiskette kann mit INIT oder COPYDISK selbst ausgeführt werden.

5.9. Diskettenformatierprogramm INIT

Aufgaben

Disketten müssen vor dem ersten Einsatz formatiert, d.h. zur Datenaufnahme bereitgemacht werden. Diese Funktion wird durch das Dienstprogramm INIT realisiert. Außerdem initialisiert INIT den Diskettenabschnitt, der von SCPX 1700 als Verzeichnisbereich genutzt wird.

Arbeitsweise

Zur Formatierung von Datenträgern werden durch SCPX 1700 keine Rufe bereitgestellt. Der Datentransport muß daher über direkte Ein-/Ausgabe erfolgen.

5.10. LDSCP-Kopierprogramm LDCOPY

Aufgaben

LDCOPY dient dem Erstellen von SCP-1700-Systemdisketten. Mit seiner Hilfe kann der SCP-1700-Steuerprogrammlader LDSCP auf die ersten beiden Spuren einer formatierten Diskette übertragen werden.

Arbeitsweise

Beim Aufruf von LDCOPY kann der Name einer CMD-Datei angegeben werden. Der Steuerprogrammlader wird dann von dort in den Hauptspeicher geladen. Ist kein Dateiname angegeben, fordert das Kopierprogramm vom Nutzer die Eingabe einer Laufwerksbezeichnung und lädt die ersten beiden Spuren der dort befindlichen Diskette. Danach erfragt LDCOPY die Laufwerksbezeichnung der Zieldiskette und überträgt den zuvor eingelesenen Speicherinhalt auf die ersten beiden Spuren dieser Diskette.

5.11. Bedienerhilfsprogramm HELP

Aufgaben

Mit HELP kann sich der Nutzer Informationen über alle möglichen SCP-1700-Kommandos anzeigen lassen.

Arbeitsweise

Das HELP-Kommando arbeitet mit einer speziellen internen Textdatei HELP.HLP.
Durch Eingabe bestimmter Suchbegriffe (topics) und diesen wiederum untergeordneten Suchbegriffen (subtopics) in den Kommandozeilen, zum Aufruf von HELP bzw. auf Anforderung von HELP, kann der Nutzer gezielt Informationen über ein bestimmtes Kommando oder einen Parameter für ein Kommando anfordern. Die Textdatei HELP.HLP kann in eine editierbare Textdatei HELP.DAT umgewandelt werden. Der Nutzer hat die Möglichkeit, diese Textdatei durch eigene Begriffe zu erweitern.

5.12. Programm zur Manipulierung des Diskettenzugriffs DISKSET

Aufgaben

In den Externspeichersteuertabellen der Executive SCPX 1700 ist für jedes der angeschlossenen Diskettenlaufwerke festgelegt, welchen physischen und logischen Aufbau die dort benutzbaren Disketten haben müssen.
Mit dem Kommando DISKSET können einige Steuertabellenwerte temporär geändert werden. Dadurch hat der Nutzer die Möglichkeit, von den SCP-Hausformaten abweichende Disketten zu verarbeiten, ohne das SCP umzugenerieren.

Arbeitsweise

DISKSET gibt dem Nutzer über Menü-Angebote vor, welche Parameter zur Auswahl stehen. Entsprechend der Nutzereingaben werden die betreffenden Werte in den BIOS-Steuertabellen geändert. Die Änderungen bleiben gültig bis zur erneuten Veränderung durch DISKSET bzw. bis zum Neustart von SCP 1700.

5.13. Steuertabellengenerierungsprogramm GENDEF

Aufgaben

GENDEF ist ein Hilfsmittel für die Anpassung von SCP 1700 an nicht standardmäßig unterstützte Externspeicher.
Insbesondere dient GENDEF zur Generierung von Steuertabellen für den Zugriff zu Foliespeichern.

Arbeitsweise

GENDEF verarbeitet eine vom Nutzer zu erstellende Quelldatei des Typs DEF und erzeugt daraus eine Assemblerquelldatei des Typs LIB.
In der DEF-Datei beschreibt der Nutzer mit vorgegebenen sprachlichen Mitteln den Typ jedes angeschlossenen Laufwerks und das zu realisierende Dateiformat. Die LIB-Datei wird so erzeugt, daß sie als INCLUDE-Datei in die Assembler-Quelldatei des BIOS übernommen werden kann.

6. Übersetzer

6.1. BASIC-Interpreter BASIC

Aufgaben

Der BASIC-Interpreter BASIC wird zum interaktiven Erstellen, Testen und Abarbeiten von BASIC-Programmen benutzt. BASIC (Beginners All-purpose Symbolic Instruction Code) ist eine in vielen Sprachversionen verbreitete, leicht erlernbare Programmiersprache, die besonders für die Lösung wissenschaftlich-technischer Aufgaben geeignet ist.

Arbeitsweise

- Interaktive Eingabe und Änderung des BASIC-Programms bzw. Eingabe eines BASIC-Programms von einer Datei
- Interpretative Abarbeitung des Programms
- Ausgabe neu erstellter oder modifizierter BASIC-Programme auf Dateien

Neben Integerzahlen und Realzahlen einfacher Genauigkeit sowie Zeichenketten werden auch Realzahlen doppelter Genauigkeit verarbeitet. Anweisungen zur sequentiellen und direkten Ein- und Ausgabe ermöglichen komfortable Dateiarbeit. Hinzu kommen Möglichkeiten der formatgesteuerten sequentiellen Ausgaben.

BASIC-Programme können durch die Möglichkeit, bis zu 32 Zeichen lange Identifikatoren für Variable und Felder zu verwenden, übersichtlich gestaltet werden.

BASIC enthält den ANSI-Standard ANSI X3.60 für Minimal-BASIC als Untermenge.
BASIC ist kompatibel zu dem auf dem Bürocomputer A5120/30 unter SCP 1520 implementierten BASIC-Interpreter BASI 1520.

6.2. FORTRAN-Compiler FOR77

Aufgaben

Die Programmiersprache FORTRAN-77 ist die aktuelle standardisierte Sprachversion von FORTRAN (ANSI X3.9-1978). FOR77 übersetzt FORTRAN-77-Quellprogramme in die Assemblersprache. Nach Assemblierung der erzeugten Assemblerprogramme (RASM86) und deren Verbindung (LINK86) mit einem FOR77-Laufzeitsystem entsteht ein ausführbares Programm.

Arbeitsweise

Der Compiler arbeitet als Mehrpaßcompiler und ist in folgende logische Stufen untergliedert:
- Initialisierung
- Syntaxanalyse
- globale Prüfung und Erzeugung von Cross-Referenzen
- Vorbereitung für die Optimierung (wahlweise)
- Ausgabe der Attribut- und Cross-Referenztabellen (wahlweise)
- Optimierung (wahlweise)
- Registerzuordnung
- Codeerzeugung

Der Compiler ist über Auswahlbedingungen steuerbar (siehe Abschnitt Leistungsfähigkeit) und ist so konzipiert, daß er an verschiedene Hauptspeichergrößen sinnvoll angepaßt werden kann. Er führt automatisch Auslagerungen von Verzeichnis- und Textblöcken auf Hilfsdateien durch, falls der zur Verfügung stehende Hauptspeicherplatz nicht ausreicht.

Leistungsfähigkeit

FOR77 realisiert den im ANSI-Standard beschriebenen Sprachumfang mit einigen anlagenspezifischen Erweiterungen (z. B. 16- und 32-bit-Integerarithmetik). FOR77 ist durch eine Reihe von Auswahlbedingungen steuerbar, die im wesentlichen folgende Funktionen erfüllen:

- Quelltextausgabe
- Unterbrechung der Compilierung in Abhängigkeit von der Fehlerschwere
- Ausgabe einer Attribut- und Cross-Referenz
- Optimierung des Zielcodes
- Auflistungssteuerung
- Festlegung von Standardlängen für INTEGER, REAL und LOGICAL
- Prüfung des Indexüberlaufs

Die Optimierung des Zielcodes enthält eine Schleifen- und eine Ausdrucksoptimierung. Durch die Schleifenoptimierung werden vor allem schleifeninvariante Codefolgen aus den Schleifen entfernt. Die Ausdrucksoptimierung erkennt gleiche Teilausdrücke (logische Ausdrücke, Adreßberechnungen, Feldzugriffe, Argumentlisten) und sichert die einmalige Berechnung.
Der Compiler gestattet eine Stapelverarbeitung von FORTRAN-77-Programmeinheiten.

7. Programmpaket für modulare Programmierung

7.1. Assembler für linkbare Programme RASM86

Aufgaben

Der Assembler übersetzt RASM86-Quellprogramme in Objektcode, der von LINK86 weiterverarbeitet wird. Im Gegensatz zu ASM86 besteht die Möglichkeit, mit Hilfe globaler Symbole Verbindungen zu anderen unabhängig übersetzten Programmen herzustellen. Für ASM86 geschriebene Quellen sind durch RASM86 assemblierbar.

Arbeitsweise

RASM86 verarbeitet eine RASM86-Assemblerquelldatei in drei Pässen und produziert daraus eine Objektcodedatei. Außerdem können zwei weitere Dateien erzeugt werden
- eine Assemblerliste mit Fehlermitteilungen
- und eine Liste der definierten Symbole.

Durch Parameter beim Aufruf von RASM86 können die Ausgabedateien unterschiedlichen logischen Geräten zugewiesen oder auch unterdrückt werden.

Der Assembliervorgang ist jederzeit durch eine Bedienereingabe unterbrechbar. RASM86 bietet Direktiven zur

- Programmsegmentierung
  Durch Angabe von Segmentparametern können dem Linker Informationen über besondere Eigenschaften des jeweiligen Segments übermittelt werden
- Definition von globalen Datenbereichen und Eintrittspunkten
- Spezifikation von externen Datenbereichen und Eintrittspunkten
- Datenstrukturierung
- Speicherplatzreservierung und -initialisierung
- Manipulation des Speicherplatzzählers
- bedingten Assemblierung
- Definition von Symbolen
- Einbeziehung von INCLUDE-Dateien
- Listensteuerung
- Verarbeitung von Code-Makros

7.2. Ausgabeprogramm für Cross-Referenzlisten XREF86

Aufgaben

XREF86 erzeugt aus der durch RASM86 erzeugten Assembler- und Symbolliste eine Cross-Referenzliste.

7.3. Bibliothekar LIB86

Aufgaben

Der Bibliothekar ermöglicht dem Nutzer die Erzeugung und Verwaltung von Objektbibliotheken, die durch den Linker beim Programmverbinden benutzt werden können.

Arbeitsweise

Die einzelnen Funktionen des Bibliothekars sind durch den Nutzer über Optionen in der Kommandozeile auswählbar. Folgende Möglichkeiten werden geboten:

- Erzeugen von Objektbibliotheken
- Löschen von Moduln aus einer Bibliothek
- Herauslösen einzelner Moduln aus einer Bibliothek
- Hinzufügen von Moduln in eine Bibliothek
- Ersetzen von Moduln in einer Bibliothek
- Herstellen einer Liste über die in einer Bibliothek enthaltenen Moduln sowie der darin definierten Segmente und globalen Symbole
- Herstellen einer Cross-Referenzliste für eine Bibliothek

Der Nutzer kann LIB86-Kommandos in eine Datei schreiben und diese Datei beim Aufruf von LIB86 als INPUT-Datei spezifizieren. Die Bibliotheksarbeit ist jederzeit durch eine Bedienereingabe unterbrechbar.

7.4. Linker LINK86

Aufgaben

Der Linker verarbeitet einen oder mehrere durch RASM86 oder Compiler für höhere Programmiersprachen erzeugte Objektmoduln zu einem verschieblichen Lademodul, der ab jeder im System möglichen 16-Byte-Grenze ladbar ist.

Arbeitsweise

Der Linker führt folgende Funktionen aus:

- Kombinieren separat erstellter Objektmoduln, dazu werden die vom Assembler bzw. Compiler weitergereichten Informationen über die in den Moduln enthaltenen Segmente ausgewertet
- Durchsuchen von Bibliotheksdateien nach Definitionen von unbekannten externen Referenzen
- Lösen externer Cross-Referenzen
- Erzeugen einer Ladedatei (CMD-Datei)
- Erzeugen einer Speicherbelegungsdatei (MAP-Datei) mit Informationen über die in der CMD-Datei enthaltenen Segmente
- Erzeugung einer Symboldatei (SYM-Datei) mit Informationen über die in der CMD-Datei enthaltenen Symbole

Die Erzeugung der CMD-, MAP- und SYM-Dateien kann durch Parameterangabe in der Kommandozeile zum Aufruf von LINK86 gesteuert werden.
Der Nutzer kann den Parameterteil für eine LINK86-Operation in eine Datei schreiben und diese Datei beim Aufruf von LINK86 als INPUT-Datei spezifizieren.

Die Arbeit des Linkers ist jederzeit durch eine Bedienereingabe unterbrechbar.

7.5. Symbolischer Debugger SID86

Aufgaben

SID86 ist ein Hilfsmittel zum Test von Nutzerprogrammen, das in erster Linie für in Assemblersprache geschriebene Programme einsetzbar ist, aber auch zur Testung von in höheren Programmiersprachen geschriebenen Programmen genutzt werden kann.

Arbeitsweise

SID86 ist eine Weiterentwicklung der Testhilfe DDT86 (siehe Abschnitt 5.5.), die es dem Nutzer gestattet, beim Testen von Programmen mit symbolischen Namen (Variablen-, Prozedur- und Marken namen) zu arbeiten. SID86 benutzt die von LINK86 erzeugte Symbolliste.

8. SCP-1700-Anpassung

Wird SCP 1700 auf einem A7100 mit nicht standardmäßig unterstützter Peripherie eingesetzt, sind Änderungen in den Gerätebedienroutinen des Steuerprogramms SCPX 1700 notwendig. Falls SCPX 1700 von einem nicht standardmäßig unterstützten Folienspeichergerät geladen werden soll, sind weiterhin Anpassungen im Steuerprogrammlader LDSCP vorzunehmen.

Im Liefersatz von SCP 1700 sind die anzupassenden Programmteile des Steuerprogrammladers (LDBIOS) und des Steuerprogramms (BIOS) als ASM86-Quellen enthalten. Zur Durchführung der Anpassung können die SCP-1700-Dienstprogramme ED, GENDEF, ASM86, PIP, GENCMD und LDCOPY genutzt werden. Der Anpassungsvorgang ist in der Anleitung für den Systemprogrammierer detailliert beschrieben.

Anlage 1 Typenbezeichnung der Geräte

Entsprechend dem derzeitigen Entwicklungsstand werden durch SCP 1700 die in Abschnitt 2. dieser Schrift genannten Geräte mit den nachfolgend aufgeführten Typenbezeichnungen bedient.

Tastatur	K 7637
Bildschirmeinheit	K 7229 (25 Zeilen, 80 Zeichen/Zeile)
Minifolienspeicher	K 5600.20 standardmäßig
Folienspeicher	K 5602.10
Seriendrucker	K 6311
Bildschirmterminal	K 8911 (anstelle von K 7637/K 7229)
Seriendrucker	SD robotron 1152-251 wahlweise
Seriendrucker	SD robotron 1157-264 wahlweise

Anlage 2 Abkürzungsverzeichnis

AC	Arbeitsplatzcomputer
ASM86	Assembler
BIOS (Basic I/O-System)	Basis-E/A-System
BDOS (Basic Disk Operating System)	Plattenbetriebssystem
CCP (Command Console Processor)	Kommandointerpreter
DDT86 (Dynamic Debugging Tool)	Testprogramm
ED	Editor
E/A	Ein- und Ausgabe
GENCMD (CMD-File Generation utility)	Ladeformatgenerierungs-programm
GENDEF (Diskdef File Generator)	Steuertabellengenerierungs-programm
IFSP	Interfaceschnittstelle parallel
IFSS	Interfaceschnittstelle seriell
KROS	Kombinat ROBOTRON Standard
LDBIOS (Loader BIOS)	Basis-E/A-System für den Steuerprogrammlader
LDSCP (SCPX-1700-Loader)	Steuerprogrammlader
LIB86 (Librerion Utility)	Bibliothekar
LINK86 (Linkage Editor)	Programmverbinder
PIP (Periphal Interchange Program)	Dateitransferprogramm
RAM (Random Access Memory)	Speicher mit wahlfreiem Zugriff
ASM86	Assembler für linkbare Programme
SCP (Single-user Control Program)	Einzelnutzerbetriebssystem
SID86 (Symbolic Instruction Debugger)	Symbolischer Debugger
STAT (Status Utility)	Systemstatusprogramm
SUBMIT (Submit Utility)	Stapelverarbeitungsprogramm

Sachwortverzeichnis

Anpassung 20, 26
Assembler 15, 23

BASIC-Interpreter 21
BDOS 11
BIOS 11
Basisseite 14
Bezugslaufwerk 13
Bibliothekar 24

CCP 11

Dateiverzeichnis 13
Dienstprogramm 7
Diskette 12

Editor 16

FORTRAN-77 21
Folienspeicher 13
Formatierung 19

Gerät 10
Gerät, logisch 11
Gerät, physisch 11

Kommando 13
Kommando, resident 13
Kommando, transient 13
Kommandodatei 18

Linker 24

Programmpaket für modulare Programmierung 7

RAM-Bereich 12

Speichermodell 14
Speicherverwaltung 12
Speicherverwaltung, dynamische 12
Steuerprogramm 7, 11
Steuerprogrammlader 7
Suchbegriff 19
Systemdiskette 19

Testhilfe 25
Transientbereich 11
Transientkommando 13f.

Übersetzer 7

Ag 706/287/87 III/12/12